Q76494.APP.CWB
July 10, 2003

INDEX STRUCTURE OF METADATA , METHOD FOR PROVIDING INDICES OF METADATA, AND METADATA SEARCHING METHOD AND APPARATUS USING THE INDICES OF METADATA

BACKGROUND OF THE INVENTION

1. Field of the Invention

[01] The present invention relates to an index structure of metadata provided for searching for information on contents and a method for providing indices of the metadata, and a method and an apparatus for searching for the metadata using the index structure of the metadata. More particularly, the present invention relates to an index structure of metadata containing information on a key, at least a part of which is encoded so as to allow information on contents to be more efficiently searched when the XML metadata for the digital contents defined in TV-Anytime Forum (hereinafter referred to as “TVA”) (hereinafter referred to as “TVA metadata”) is divided into fragments in an independent unit and transmitted on a fragment basis, a method for providing indices of the metadata, and a method and an apparatus for searching the metadata using the indices of metadata. The present application is based on Korean Patent Application Nos. 2002-43097 and 2002-62913, which are incorporated herein by reference.

2. Description of the Prior Art

[02] The TV-Anytime Forum is a private standardization organization established in September 1999 with the purpose of developing standards for providing audiovisual related services in a user-friendly environment such as a personal digital recorder (PDR) having a high volume personal storage device. Specifically, the aim of the services is to enable all the users to view and listen to various types of programs (such as conventional broadcasting services, online interactive services and the like) at a desired time and in a desired manner based on the personal storage device.

[03] The TV-Anytime Forum has operated Working Groups for business models, system/transmission interfaces/contents referencing, descriptions, metadata, rights management and protection and the like, in order to establish standardization. With respect to the metadata concerned in the present invention, "1st Draft of Metadata Specification SP003v1.3" up to June 2002 has been published.

[04] A configuration of the PDR will be briefly described with reference to FIG. 1. The PDR 100 receives video/audio signals and metadata via a variety of networks such as sky waves, satellite waves, internet networks and the like from a provider 200 for providing video/audio signals, collects viewing and listening patterns, and personal tastes of users, if necessary, and transmits them to the provider 200 for providing the video/audio signals. The PDR 100 comprises a high volume storage device for storing therein the received

video/audio signals and metadata. The PDR 100 further comprises software for storage and reproduction of the video/audio signals, and an electronic program guide (EPG) application for retrieving and displaying metadata for the video/audio signals. The user ascertains the metadata for the video/audio data, i.e., titles of the programs, program reproduction times and the like, through a grid guide screen of the EPG application shown in FIG. 2, selects a desired program, and receives it via the network in real time or reproduces the video/audio data previously stored in the high volume storage device.

[05] The metadata refer to data describing contents such as titles and synopses of programs, and are defined as "data about data." In the TVA metadata specifications of the TV-Anytime Forum, its structure is defined by use of XML schema language (see XML 1.0 of W3C), the standard by the W3C (a consortium for promoting standards for the XML), and the semantics and attributes of the respective metadata elements are also defined. The TVA metadata relevant to broadcasting contents are configured with an XML document having a root node, "TVAMain (300)" as shown in FIG. 3. The TVA metadata relevant to programs are configured with, for example, nodes such as ProgramInformation Table, GroupInformation Table, ProgramLocation Table, ServiceInformation Table and the like, under the node of "ProgramDescription."

[06] In the TV-Anytime Forum, the TVA metadata are transmitted on a fragment basis as an independent unit in order to transmit a large volume of

TVA metadata in a stream format. The concept of fragments will be briefly described with reference to FIG. 4. The fragments are obtained by dividing the TVA metadata configured with the XML documents shown in FIG. 3 into predetermined tree structures. For example, where the entire TVA metadata are divided into a tree structure (fragment TVAMain) including an upper node of "TVAMain" and predetermined child nodes under this upper node, a tree structure (fragment ProgramInformation) including an upper node of ProgramInformation Table and child nodes under this upper node, a tree structure (fragment BroadcastEvent) including an upper node of the BroadcastEvent Information and child nodes under this upper node, each of the divided tree structures becomes a fragment. The fragments can be transmitted independently of the other fragments, and the fragments can be accessed individually.

[07] For individual access to the fragments, it is necessary to know a node referenced by a transmitted TVA metadata fragment, i.e., a node corresponding to the upper node of the TVA metadata fragment, in the entire metadata tree structure, and to describe relative paths in the TVA metadata fragments of keys contained in the transmitted TVA metadata fragment. To this end, XPath, which is a syntax for describing a path to one or more nodes in an XML document defined by W3C, is used. The term 'key' refers to a specific field of the metadata used for indexing, and also means child nodes of a node referenced by a fragment. Fields (for search conditions) input by the user, such as 'Service ID' and 'Published Time,' correspond to the keys.

[08] In order to provide efficient search for and access to fragments, an index structure for the keys included in the metadata fragments is additionally required, and information on the index structure, i.e., index information, is also transmitted independently of the metadata fragments.

[09] Under the environment provided by the TV-Anytime Forum, if a user desires to retrieve information on a program meeting a predetermined Published Time condition, the index information transmitted thereto independently of the fragments is utilized to identify the location (identifier) of a metadata fragment meeting a desired Published Time condition and an access to the relevant metadata fragment is then made based on the location (identifier), so as to extract metadata meeting the Published Time condition.

[10] TV-Anytime Specification TV145, J.P. Evain, "1st Draft of Metadata Specification SP003v1.3", TV-Anytime Forum 17th meeting, Montreal, Canada, June 2002; hereinafter, referred to as "Key index art reference" proposes a key index data stream structure for a metadata fragment index.

[11] The notion of a container defined by the TV-Anytime Forum will be described prior to describing the index structure.

[12] The TV-Anytime Forum defines a container as a top-level storage to which all the data covering the aforementioned index information and the metadata fragments are transmitted, which is called a type of top-level transmission. Describing the container briefly, each container comprises a plurality of sections, each storing therein the index information or the metadata fragments. The container can be classified into an index container and a data container according to the information carried thereby: the index container carries index information sections such as a key index list (key_index_list) section, a key index (key_index) section, a sub key index (sub_key_index) section, a string repository (string_repository) section and a fragment data repository (fragment_data_repository) section, whereas a data container carries metadata fragment sections such as an elements table (elements_table) section, a string repository (string_repository) section and a fragment data repository (fragment_data_repository) section. The above classification is done based on the contents of the information included in the containers. Both the index container and the data container are identical in configuration.

[13] Referring to the container defined by the TV-Anytime Forum as illustrated in FIG. 5, the container comprises a container identifier (container_id) data field (not shown) and a large number of sections. In each section, the contents stored in 'section_body' are identified according to an encoded value in 'section_id'. For example, a section 10 of which the encoded value in 'section_id' is '0X0004' is identified as a key index list (key_index_list) section, a section 20 of which the encoded value in 'section_id' is '0X0005' is identified as a key index (key_index) section, a section 30 of which the encoded value in ‘section id’ is ‘0X0006’ is identified as a sub key index (sub_key_index) section, a section 40 of which the encoded value in ‘section id’ is ‘0X0001’ is identified as an element table (element_table) section, and a section 50 of which the encoded value in ‘section id’ is ‘0X0003’ is identified as a fragment data repository (fragment_data_repository) section.

[14] The TVA metadata fragments are stored in the fragment data repository (fragment_data_repository) section 50 of the data container and then transmitted. The identifier information (handle_value) for the TVA metadata fragments in the data container is included in the element table section 40 of the data container.

[15] In conclusion, the TVA metadata fragment is uniquely identified by the container identifier information (container_id) and the metadata fragment identifier information (handle_value) of the container that includes the TVA metadata fragment.

[16] The key index art reference described above proposes the key index structure for indexing the TVA metadata fragments stored in the aforementioned data container, i.e., a structure composed of the key index list (key_index_list) section 10, the key index (key_index) section 20, and the sub key index (sub_key_index) section 30. Since the syntax of the structure is described in detail in the key index art reference described above, the detailed description thereof will be omitted. Hereinafter, the structure will be

described with reference to FIG. 6 that illustrates the structure by segments of the index information.

[17] The key index list (key_index_list) section 10 defined in the key index structure provides a list of all the keys transmitted. The list includes key information defining each key and identification information on the key index (key_index) section 20 to be described later. The key information comprises (1) location information of the metadata fragment relevant to the key, (2) location information of the key within the metadata fragment and identification information for the key index (key_index) section 20 which will be set forth later. The location information of the metadata fragment is expressed in XPath (fragment_xpath_ptr) in the TVA. The location information of the key is expressed in XPath (key_xpath_ptr) for the relative path within the relevant fragment of the nodes used as the key in the TVA.

[18] The XPath of the metadata fragment is a path to the root node of the TVA metadata XML document, i.e., an absolute path, and the XPath of the nodes used as the keys, i.e., the XPath of the keys, represents a relative path of the key for the relevant metadata fragment. The XPath for the metadata fragment and the XPath for the key are stored in a 'fragment_xpath_ptr' segment 11 and a 'key_xpath_ptr' segment 12, respectively.

[19] Furthermore, the key index list (key_index_list) section 10 includes the identification information on the key index (key_index) section 20 of each key to be described later (i.e., the container identifier information (container_id) of

the container storing therein the key index (key_index) section 20 and the key index identifier information). The container identifier information and the key index identifier information are stored in an 'index_container' segment of the key index list (key_index_list) section 10 and a 'key_index_identifier' segment, respectively, and then transmitted.

[20] The key index (key_index) section 20 defined in the key index structure provides a list of all the sub key index (sub_key_index) sections 30 to be described later. The list includes information representing the ranges of values of the key included in the respective sub key index (sub_key_index) sections 30, i.e., the highest value of the key among the values of the key within each sub key index (sub_key_index) section 30 (hereinafter referred to as 'representative key value'), and identification information on the sub key index (sub_key_index) section 30 relevant to each representative key value (i.e., the container identifier information (container_id) of the container storing therein the sub key index (sub_key_index) section, and the sub key index identifier information).

[21] Accordingly, the key index section (key_index) 20 includes a 'key_index_identifier' segment for storing therein the key index identifier information defined in the key index list (key_index_list) section 10, a 'high_key_value' segment 13 for storing therein the representative key values of the respective sub key index (sub_key_index) sections 30, the container identifier information (container_id) of the container in which the sub key index (sub_key_index) section 30 is stored, a 'sub_index_container' segment for storing respective sub key index identifier information and a 'sub_index_identifier' segment. The sub key index (sub_key_index) section 30 defined in the key index structure provides a list of the values of the key included in the relevant sub key index (sub_key_index) section 30. The list includes the values of the key included in the relevant sub key index (sub_key_index) section 30 and the identification information on the metadata fragments having the values of the key (i.e., the container identifier information (container_id) of the containers storing the metadata fragments and the identifier information (handle_value) of the metadata fragments).

[22] Accordingly, the sub key index (sub_key_index) section 30 includes a 'sub_index_identifier' segment for storing therein the sub key index identifier information defined in the key index (key_index) section 20, a 'key_value' segment 14 for storing therein the values of the key, a 'target_container' segment for storing therein the container identifier information (container_id) of the containers in which the metadata fragments are stored, and a 'target_handle' segment for storing therein the fragment data identifier information (handle_value). The key index structure may be more easily understood by referring to FIG. 7 illustrating the index information.

[23] FIG. 7 shows the key index list (key_index_list) section including keys relevant to the Service Id, the Published Time and the Published Duration. The upper node of the metadata fragment including the keys relevant to the

Service Id, the Published Time and the Published Duration is 'BroadcastEvent' 310 as shown in FIG. 3, identified by a shaded block. Accordingly, the XPath '/TVAMain/ProgramDescription/ProgramLocation Table/BroadcastEvent' for the 'BroadcastEvent' fragment is stored in the 'fragment_xpath_ptr' segment 11a, and the XPaths to the keys of the Service Id, the Published Time and the Published Duration for the 'BroadcastEvent' fragment, i.e., '@ServiceId' (311a in FIG. 3), 'EventDescription/ PublishedTime' (311b in FIG. 3) and 'EventDescription/PublishedDuration' (311c in FIG. 3) are stored in the 'key_xpath_ptr' segment 12a.

[24] The index structure will be more comprehensible with reference to FIG. 7 which illustrates the index information.

[25] FIG. 7 shows the key index list (key_index_list) section including keys for Service ID, Published Time and Published Duration, wherein a upper node of the metadata related to the Service ID, the Published Time and the Published Duration is 'BroadcastEvent' 310 indicated as a shaded portion in FIG. 3. Accordingly, the XPath for the 'BroadcastEvent' fragment, '/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent' is stored in the 'fragment_xpath_ptr' segment, and the respective XPaths for keys of Service ID, Published Time and Published Duration for the 'BroadcastEvent' fragment, '@ServiceID' (see 311a of FIG. 3), 'EventDescription/PublishedTime' (see 311b of FIG. 3), and

'EventDescription/PublishedDuration' (see 311c of FIG. 3) are stored in the 'key_xpath_ptr' segment.

[26] Also, FIG. 7 shows the key index (key_index) section 20 and the sub key index (sub_key_index) section 30 for the Service ID (the XPath of the key: @ServiceID) among the key index list (key_index_list) sections.

[27] In such an index structure, when search conditions for searching the metadata are input, location information on the fields of the input search conditions in the metadata is determined and the determined location information is compared to the key information in the key index list so as to search the key having the determined location information within the key index list, overhead is caused since comparison of both Xpaths is necessary. The same problem occurs when the keys indicating relative paths from the fragments among the key information are compared in terms of location information. Particularly, this problem becomes more severe when fragments, which are more complex than the keys, are compared in terms of location information. Since the XPath of the fragment representing location information among key information describes a path to a relevant node from the root node on the XML document, transmission costs are inefficient and interpretation costs of the XPath in the terminal are high. For example, the XPath of the broadcast event fragment indicating location information of a program among the TV-Anytime fragments can be expressed as '/TVAMain/ProgramDescription/ProgramLocationTable/BroadcastEvent'.

Meanwhile, in order to represent one node on the XML document, the XPath can be expressed in an alternative manner. In the case of a broadcast event, in addition to the aforementioned normal representation, the XPath can be expressed alternatively, such as '/TVAMain//BroadcastEvent' or '//BroadcastEvent,' and so on. Herein, '//' means a child node in the structure of an XML document. Therefore, an operation to inspect whether fragments are the same by use of the XPath is not a simple one that merely matches simple strings with each other. In particular, overhead is caused in analysis/comparison of the relevant path, if the XPath path is expressed in an abbreviated format.

SUMMARY OF THE INVENTION

[28] The present invention is contemplated to solve the aforementioned problems. An object of the present invention is to provide an index structure of metadata including information of a key encoded so as to allow information on contents to be searched more quickly.

[29] Another object of the present invention is to provide a method of providing an index of the metadata capable of searching the information on contents in a fast manner, a method of searching the metadata using the metadata index, and a searching apparatus using the same.

[30] According to one embodiment of the present invention to accomplish these and other objects, there is provided an index structure of metadata comprising a list of keys composed of predetermined fields of the metadata, wherein the list contains therein location information of the fields in the metadata, wherein at least a part of the location information is expressed as a predetermined code.

[31] Preferably, the index structure further comprises values of the key and identification information of the metadata corresponding to the values of the key.

[32] Also preferably, the metadata comprises fragments divided by a predetermined range in a tree data structure, wherein the field constituting the key corresponds to any one of the information constituting the fragments.

[33] It is desirable that the identification information of the metadata comprises identification information of the fragment.

[34] Desirably, the location information comprises location information of the fragment to which the field constituting the key belongs within the data structure and location information of the field within the fragment.

[35] Desirably, either the location information within the data structure or the location information within the fragment is expressed in a predetermined code.

[36] It is preferable that at least a part of the location information is expressed in XPath.

[37] Preferably, the code is assigned in advance to the location information frequently referred.

[38] More preferably, the index structure further comprises a representative key value representing the predetermined range of the values of the key.

[39] Desirably, the representative key value comprises at least one of a maximum value, a minimum value or an intermediate value among the values within the concerned range.

[40] It is desirable that the metadata has a structure of metadata as defined in TVA.

[41] According to one embodiment to accomplish these and other objects of the present invention, there is provided a method for providing the metadata index including a list of the keys composed of predetermined fields of the metadata, wherein the list contains location information of the field in the metadata, wherein at least a part of the location information is expressed with a predetermined code.

[42] Preferably, the metadata index further comprises values of the key and identification information of the metadata corresponding to the values of the key.

[43] Desirably, the metadata comprises fragments divided by a predetermined range in a tree data structure, wherein the field constituting the key corresponds to any one of the information constituting the fragments.

[44] It is preferable that the identification information of the metadata comprises identification information of the fragment.

[45] It is also preferable that the location information comprises location information of the fragment to which the field constituting the key belongs within the data structure and location information of the field within the fragment.

[46] Preferably, either the location information within the data structure or the location information within the fragment is expressed in a predetermined code.

[47] Also preferably, at least a part of the location information is expressed in XPath.

[48] Preferably, the code is assigned in advance to the location information frequently used.

[49] Desirably, the metadata index further comprises a representative key value representing the predetermined range of the values of the key.

[50] Further desirably, the representative key value comprises at least one of a maximum value, a minimum value or an intermediate value among the values within the concerned range.

[51] Also desirably, the metadata has a structure of metadata as defined in TVA.

[52] According to one embodiment to accomplish the present invention, there is also provided a method of searching the metadata, comprising the steps of (i) determining location information of the field of the search conditions input by the user, in the metadata; (ii) searching the key containing the code predetermined as location information, where at least a part of the location information is defined as the predetermined code; and (iii) extracting the concerned metadata by use of the searched key.

[53] It is desirable that the metadata index comprises values of the key and identification information of the metadata corresponding to the values of the key.

[54] It is also desirable that the metadata comprises fragments divided by a predetermined range in a tree data structure, wherein the field constituting the key corresponds to any one of the information constituting the fragments.

[55] Desirably, the identification information of the metadata comprises identification information of the fragment.

[56] Also desirably, the location information comprises location information of the fragment to which the field constituting the key belongs within the data structure and location information of the field within the fragment.

[57] Preferably, either the location information within the data structure or the location information within the fragment is expressed in a predetermined code.

[58] Also preferably, at least a part of the location information is expressed in XPath.

[59] It is preferable that the code is assigned in advance to the location information frequently used.

[60] It is also preferable that the metadata index further comprises a list of the keys.

[61] Also preferably, the metadata index further comprises the representative key value representing the predetermined range of the values of the key.

[62] Preferably, the representative key value comprises at least one of a maximum value, a minimum value or an intermediate value among the values within the concerned range.

[63] Further preferably, the metadata has a structure of metadata as defined in TVA.

[64] It is desirable that the step (ii) of searching the key comprises the step of searching the key containing the code defined as location information in the key list where (a) location information in the data structure or (b) location information in the fragment is defined with a predetermined code.

[65] Desirably, the step (iii) of extracting the metadata comprises the steps of (iii-1) searching a value of the key meeting the input search conditions among the values of the key to be indexed by the searched key, and (iii-2) extracting the concerned metadata by use of the searched value of the key.

[66] Desirably, the step (iii-1) of searching a value of the key meeting the input search conditions among the values of the key to be indexed by the searched key comprises the steps of searching the representative key value meeting the input search conditions, and searching the value of the key meeting the input search conditions among the values of the key in the range represented by the representative key value.

[67] According to one embodiment to accomplish these and other objects of the present invention, there is provided an apparatus for searching metadata, comprising an input unit allowing a user to input search conditions, and a control unit determining location information of the field of the search conditions input by the user, in the metadata, searching the key containing the code predetermined as location information, where at least a part of the location information is defined as the predetermined code, and extracting the concerned metadata by use of the searched key.

[68] Preferably, the metadata index comprises values of the key and identification information of the metadata corresponding to the values of the key.

[69] Preferably, the metadata comprises fragments divided by a predetermined range in a tree data structure, wherein the field constituting the key corresponds to any one of the information constituting the fragments.

[70] Preferably, the identification information of the metadata comprises identification information of the fragment.

[71] Preferably, the location information comprises location information of the fragment to which the field constituting the key belongs within the data structure and location information of the field within the fragment.

[72] Preferably, either the location information within the data structure or the location information within the fragment is expressed in a predetermined code.

[73] Preferably, at least a part of the location information is expressed in XPath.

[74] Preferably, the code is assigned in advance to the location information frequently used.

[75] Preferably, the metadata index further comprises a list of the keys.

[76] Preferably, the metadata index further comprises the representative key value representing the predetermined range of the values of the key.

[77] Preferably, the representative key value comprises at least one of a maximum value, a minimum value or an intermediate value among the values within the concerned range.

[78] Preferably, the metadata has a structure of metadata as defined in TVA.

[79] Preferably, the control unit searches the key containing the code defined as location information, in the key list, where (a) location information in the data structure or (b) location information in the fragment is defined with a predetermined code.

[80] Preferably, the control unit searches the value of the key meeting the input search conditions among the values of the key to be indexed by the searched key, and extracts the concerned metadata by use of the value of the searched key.

[81] Preferably, the control unit searches the representative key value meeting the input search conditions, and searches the value of the key meeting the input search conditions among the values of the key in the range represented by the representative key value.

[82] Also preferably, the searching apparatus further comprises a receiving unit receiving metadata, a storage unit storing therein the received metadata, and an output unit outputting the search result by the control unit.

[83] Therefore, an apparatus for searching metadata that searches the TVA metadata can more efficiently perform searches for the metadata fragments by using the encoded key information.

BRIEF DESCRIPTION OF THE DRAWINGS

[84] The above and other objects and features of the present invention will become apparent from the following description of preferred embodiments given in conjunction with the accompanying drawings, in which:

[85] FIG. 1 is a schematic diagram illustrating a concept of a general PDR;

[86] FIG. 2 shows a grid guide screen in a general EPG application;

[87] FIG. 3 shows a structure of general metadata defined by the TV-Anytime Forum;

[88] FIG. 4 is a schematic diagram illustrating a concept of a general fragment defined by the TV-Anytime Forum;

[89] FIG. 5 is a schematic diagram illustrating a concept of a general container defined by the TV-Anytime Forum;

[90] FIG. 6 shows an index structure of metadata using the conventional key scheme;

[91] FIG. 7 illustrates an index structure of metadata and a searching process using the conventional key scheme;

[92] FIG. 8 shows an index structure of metadata according to an embodiment of the present invention;

[93] FIG. 9 shows an index structure of metadata and a searching process according to an embodiment of the present invention;

[94] FIG. 10 illustrates a method of providing indices of metadata according to an embodiment of the present invention;

[95] FIG. 11 is a diagram showing a method of searching for the metadata according to an embodiment of the present invention; and

[96] FIG. 12 is a schematic diagram illustrating an apparatus for searching for the metadata according to an embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[97] Hereinafter, an index structure of metadata provided for searching for information on contents, and a method for providing indices of the metadata, and a method and an apparatus for searching for the metadata using the index structure of the metadata will be described in detail with reference to the accompanying drawings.

[98] The embodiments will be described on the basis of TVA metadata in this specification for the sake of description; however, this will not be interpreted or comprehended in limiting the coverage of protection of the present invention.

[99] First, as information on contents, an index structure of metadata for searching the metadata, the syntax defining the index structure including information of an encoded key so as to index TVA metadata fragments stored in the data container as described above. That is, a key index list (key_index_list) section 110, a key index (key_index) section 120 and a sub key index (sub_key_index) section 130 will be described, and the index structure including the encoded key information defined by the syntax will be then described.

[100] The syntax defining the index structure of the metadata according to one embodiment of the present invention, in particular, including the encoded key information, is different in concept from the syntax defined in a conventional key index art reference in that it comprises structures newly introduced for an encoding concept of the key information, such as fragment_descriptor() and key_descriptor(), and reorganizes structures of the key index list (key_index_list) section 110, the key index (key_index) section 120 and the sub key index (sub_key_index) section 130.

[101] The key index list (key_index_list) section 110 as described above comprises key information defining respective keys and identification information on the key index (key_index) section 120 to be described later.

[102] The key information serves to define the key, i.e., location information in the metadata, which predetermined fields of the metadata constituting the keys have. The key information comprises location information that the metadata fragment to which the fields constituting the keys belong within the metadata (hereinafter referred to as "location information of fragment," which is expressed in XPath of the fragment in TVR (fragment_xpath_ptr)), and location information that the fields constituting the keys have within each metadata fragment (hereinafter referred to as "location information of key, that is, an XPath for relative path within the relevant fragment of the node used as the key in TVA, being expressed in XPath of the key (i.e., key_xpath_ptr).

1. Key Index List (key_index_list) section

[103] The key index list (key_index_list) section provides a list of all the transmitted keys.

[104] 'fragment_xpath_ptr' indicating location information of the fragment within the conventional key index list (key_index_list) section (expressed in XPath of fragment in the TVA) is replaced with fragment_descriptor().

Table 1

Syntax	**No. of Bits (changeable)**
key_index_list() {	
for (j=0; j<key_index_count; j++) {	
fragment_descriptor()	16
key_descriptor()	16
index_container	16
key_index_identifier	8
}	
}	

[105] **key_index_count:** specifies the number of all the transmitted keys, i.e., the number of indices for the entire XML document.

[106] **fragment_descriptor()**: describes XPath location of target fragments to be indexed. Where the location information of the fragment is expressed as predetermined codes , the same type of fragment as the standard fragment type in Table 3, to be shown later, can be described. The type of the fragment is not limited to the standard fragment type of Table 3, and the fragment can be shaped as random as possible as far as its shape can indicate XPath of the fragment to define the keys (for example, a part may be Xpath but the other part may have encoded code values).

[107] **key_descriptor()**: describes XPath of the key within the XPath location of the group of the target fragments to be indexed. Where the location information of the key is expressed as predetermined codes , the same type of fragment as the standard key type can be described. As described above with reference to the fragment_descriptor(), the type of the key is not limited to the standard key type.

[108] **index_container**: identifies the container in which a specified key index (key_index) section exists.

[109] **key_index_identifier**: identifies the key index (key_index) section within the container specified by index_container. The key index (key_index) section can be identified in a unique manner in combination of the index_container and the key_index_identifier.

2. Fragment Descriptor (fragment_descriptor)

[110] 'fragment_descriptor()' provides a structure of encoding specific bits (which may be encoded to arbitrary bits such as 8 bits, 16 bits and so on) relative to the standard fragment type frequently used, and at the same time, a structure capable of describing XPath as additional information relative to the metadata fragment type defined by the user. That is, where fragment_descriptor is '0xFF', it indicates a user-defined fragment, and thus, XPath for the relevant user-defined fragment is immediately described.

Table 2

Syntax	**(No. of Bits changeable)**
fragment_descriptor() {	
fragment_type	8

if (fragment_type == 0xFF)
{	

fragment_xpath_ptr	16
}	
}	

[111] **fragment_type:** represents the type of fragments to be indexed. Encoded values are assigned to standard fragment types frequently used. If fragment_type has an encoded value of 0xFF, fragment_xpath_ptr is added as additional information.

[112] Table 3 illustrates encoded values for location information of the frequently used fragment types (hereinafter referred to as 'standard fragment') when a search is conducted in the TV-Anytime. However, the standard fragment types and the encoded values in this embodiment are not limited to those illustrated in Table 3 but can be extended in accordance with applications.

Table 3

Value	Description
0x00	Not Designated
0x01	ProgramInformation fragment
0x02	GroupInformation fragment
0x03	CreditsInformation fragment
0x04	ProgramReview fragment
0x05	SegmentInformation fragment
0x06	ServiceInformation fragment
0x07	BroadcastEvent fragment
0xFF	User deisgnated fragment
0x08-0x0E 0x10-0xFF	Reserved

3. Key Descriptor (key_descriptor)

[113] 'key_descriptor()' provides a structure of encoding location information of the keys having a high frequency of use (hereinafter referred to as "standard key") to specific bits when a search is made, and at the same time, a structure of describing the key type defined by the user in XPath. For example, if key_descriptor is '0xFF', it indicates a user-defined key. Thus, the XPath is described as additional information for the user-defined key.

Table 4

Syntax	**No. of Bits (changeable)**
key_descriptor() {	
key_type	8
if (key_type = = 0xFF) {	
key_xpath_ptr	16
}	
}	

[114] **key_type**: represents the type of keys to be indexed. Encoded values are assigned to location information of the standard key types frequently used when a search is conducted. If the key_type has an encoded value of '0xFF', the key_xpath_ptr is added as additional information.

[115] **key_xpath_ptr**: refers to the relative path involved in the fragment XPath of the node used as the key.

[116] In the present embodiment, although the encoded values for the standard keys have not been specified, it will be understood that the encoded values for the standard key types have a structure similar to encoding of the fragment types of Table 3.

4. Key Index (key_index) section

[117] Since the definitions of the key index (key_index) section and the sub key index (sub_key_index) section are the same as those defined in the key index art reference, the detailed description thereof will be omitted.

Table 5

Syntax	No. of Bits (changeable)
key_index() {	
key_index_identifier	8
for (j=0; j<sub_index_count; j++) {	
high_key_value	16
sub_index_container	16
sub_index_identifier	8
}	
}	

5. Sub Key Index (sub_key_index) section

Table 6

Syntax	No. of Bits (changeable)
sub_key_index() {	
sub_index_identifier	8

for (j=0; j<reference_count; j++)
{	

key_value	16
target_container	16
target_handle	16
}	
}	

[118] Hereinafter, the metadata structure defined by the syntax described above will be discussed with reference to FIG. 8, in which the metadata is expressed as segments of the index information.

[119] The key index list (key_index_list) section 110 defined in the index structure provides a list of all the transmitted keys. The list includes key information defining each key (i.e., location information of the fragment (fragment_descriptor) and/or location information of the key (key_descriptor); the location information of the fragment or the location information of the key may be selectively encoded, or they may be encoded simultaneously depending on embodiments of the present invention) and identification information on the key index (key_index) section 120 to be described later. The XPath of the metadata fragment is a path for the root node of the TVA metadata XML document, i.e., an absolute path, in the same manner as in the conventional index structure, and the XPath of the node used as the key, i.e., the XPath of the key, represents a relative path of the key for the metadata fragment. The XPath of the metadata fragment and the XPath of the key in combination represents location information of the key for the entire XML document.

[120] In the present invention, the encoded value of the XPath for the metadata fragment (that is, location information of the fragment group) and the encoded value of the XPath of the key (that is, location information of the key) are respectively stored in the 'fragment_descriptor' segment 111 and the 'key_descriptor' segment 112.

[121] As describe above, where location information of the fragment among the key information is of the standard fragment type which is frequently used, there is provided an encoded value (fragment_descriptor) expressing the XPath for the metadata fragment (fragment_xpath_ptr) with predetermined codes. As the standard fragment types frequently used, there are for example, program information (ProgramInformation), program group information (GroupInformation), credit information (CreditInformation), program review (ProgramReview), segment information (SegmentInformation), broadcast event (BroadcastEvent), service information (ServiceInformation) and the like. If the XPath of the metadata fragment for these fragment types can be simply expressed as an encoded value, the overhead in the search for the metadata can be reduced.

[122] Therefore, in the index structure according to the present embodiment, the XPath of the standard metadata fragment is encoded to a predetermined encoded value and then stored. Furthermore, all of the encoded values are not assigned to the fragments and some of the encoded values (e.g., '0XFF') are assigned to the metadata fragments as defined by the user, to thereby allow the user to additionally define location information on the metadata fragment by means of the XPath. In this regard, an additional area ('fragment_xpath_ptr'), for example, by which the XPath for the metadata fragment can be designated is provided.

[123] In the embodiment in which fragments are encoded in accordance with Table 3, the location information on the metadata fragment among the key information has such encoded values as '0x01', '0x02' and '0x03.' The location information on the metadata fragment encoded to '0x01' indicates the XPath of the 'program information (ProgramInformation) fragment.' Further, where the location information on the metadata fragment is '0xFF,' it means the metadata fragment defined by the user, and thus, an additional area for enabling the XPath of the metadata fragment to be designated is provided.

[124] Although the above embodiment has been described with respect only to the metadata fragment, the same will be applied with respect to the key for the metadata fragment. As the frequently used key, encoded values can be designated for use, but the conventional XPath of the key cannot be used. In addition, if the encoded value comprises a predetermined value, the user can additionally designate the XPath for the key. The encoding of the XPath of the aforementioned metadata fragment and the encoding of the XPath of the key can be used simultaneously or independently.

[125] Furthermore, the key index list (key_index_list) section 110 comprises the identification information on the key index (key_index) section 120 of each key to be described later (i.e., the container identifier information (container_id) of the container storing therein the key index (key_index) section 120, and the key index identifier information). The container identifier information and the key index identifier information are respectively stored in an 'index_container' segment and a 'key_index_identifier' segment in the key index list (key_index_list) section 110.

[126] Since the key index (key_index) section 120 and the sub key index (sub_key_index) section 130 are the same as described in the key index art reference, the description thereof will be omitted.

[127] The index structure including the encoded key information will be described in detail with reference to FIG. 9, which illustrates the index information.

[128] FIG. 9 shows the key index list section 110 in which the XPath of 'BroadcastEvent' fragment for the Service Id is encoded to '0x07.' Herein, the key index (key_index) section 120 and the sub key index (key_index) section 130 are the same as described with reference to FIG. 7.

[129] The index structure described above is very effective when the keys related to the frequently used fragments types, e.g., ProgramInformation, GroupInformation, and BroadcastEvent and so on are used, thereby reducing the entire overhead in the apparatus for searching metadata.

[130] FIG. 10 illustrates a method of providing an index of metadata having a structure according to one embodiment of the present invention as described above.

[131] Indices of the metadata according to one embodiment of the present invention can be generated by the provider 200 providing, for example, audio/visual signals.

[132] Information on contents, that is, metadata, is first processed on a fragment basis as described above (S100). At least a portion (location information of the fragment or location information of the key) of information on the fields that will be included in the metadata index, that is, information on the key (for example, location information of the fragment and location information of the key) is encoded (S200). In other words, where location information of the metadata fragment to which fields constituting the key belong or location information of the key is of the standard fragment type or the standard key type, both of which can be encoded, the location information of the metadata fragment or the location information of the key, i.e., the XPath of the metadata fragment or the XPath of the key is encoded to the predetermined code value (for example, the 'broadcast event (BroadcastEvent) fragment is encoded to '0X07') in FIG. 9. Where the location information of the metadata fragment or the location information of the key is not identified by the encoded values, the key information expressed with XPath is designated as in the conventional art.

[133] A key is provided by use of information constituting the fragments, for example, information on 'Service ID' (S300). Then, sub key index (sub_key_index) sections 114 are provided by key as provided above (S400). The sub key index (sub_key_index) sections 114 include therein the values of the keys divided by a predetermined range, whereas the sub key index (sub_key_index) sections 114 include therein metadata fragment identification information corresponding to the values of the keys (that is, the container identifier information (container_id) and fragment data identifier information (handle_value) respectively stored in the 'target_container' segment and the 'target_container' segment of FIG. 8).

[134] The key index (key_index) section 120 containing the representative key value representing the values of the keys divided by a predetermined range is provided (S500). For example, the representative key value (e.g., 509) indicating the predetermined range (e.g., 500~509) of the Service Id is included. The key index (key_index) section 120 includes therein identification information for sub key index (sub_key_index) sections 114a and 114b storing therein the values of the keys divided by the predetermined range, wherein the identification information comprises the container identifier information (container_id) of the container in which the sub key index (sub_key_index) section is stored and the sub key index identifier information as shown in FIG. 8.

[135] The key index list (key_index_list) section 110 arranging key information as provided above, that is, location information of the fragment and location information of the key, based on the key, is provided (S600). At this time, if the encoded location information of the fragment or the encoded location information of the key in the step of S200 exists, the location information above is expressed as encoded codes when the key index list (key_index-list) section 110 is provided. In other words, the 'broadcast event (BroadcastEvent)' fragment in FIG. 9 is expressed as '0X07.' Where the location information of the fragment or the location information of the key can not be distinguished by encoded values, the key information expressed in XPath as in the conventional art is inserted.

[136] The key index list (key_index_list) section 110 further comprises identification information on the key index (key_index) section 120, in addition to the key information.

[137] The steps described above may proceed in reverse order in other embodiments, and the step S500 of providing the key index (key_index)

section 120 including therein the representative key value may be omitted depending on the embodiment(s).

[138] Hereinbelow, a method of searching for metadata meeting search conditions by use of the metadata index having a structure according to one embodiment of the present invention described above will be described with reference to FIG. 11.

[139] Search conditions are input by the user (S1100), and location information in the metadata relative to the field of the input search conditions is determined (S1200). Where at least a part of the location information, e.g., location information of the fragment or location information of the key, is defined with predetermined codes, the key including the defined key therein is searched in the key index list (key_index_list) section 110 (S1200), and the concerned metadata is extracted by use of the searched key (S1400).

[140] The step of extracting the concerned metadata, S1400, comprises the steps of searching the representative key value meeting the search conditions in comparison of the representative key value and the range of the values of the key of the search conditions in the key index (key_index) section 120, and searching the sub key index (sub_key_index) section 114 including the value of the key in the range represented by the searched representative key value (S1410), searching the value of the key meeting the search conditions in the searched sub key index (sub_key_index) section 114, and extracting the concerned metadata by use of the identification information of the metadata fragment corresponding to the value of the key by use of the searched value of the key, whereby the metadata meeting the search conditions is extracted.

[141] Here, the location information of the fragment refers to an absolute path of the metadata fragment, the keys of which are to be indexed as described above, that is, the XPath of the metadata fragment (fragment_xpath_ptr), and the location information of the key refers to a relative path of the key for the metadata fragment (relative path in the XPath location of the fragment), that is, the XPath (key_descriptor) of the nodes used as keys.

[142] In the steps of S1410, S1420 and S1430, the steps of searching the concerned key index (key_index) section 120 and the sub key index (sub_key_index) section 114, and extracting the concerned fragment proceed by use of the identification information of the key index (key_index) section 120, of the sub key index (sub_key-index) section and of the metadata fragment, respectively.

[143] FIG. 12 depicts an apparatus for searching the metadata according to one embodiment of the present invention. The apparatus performs a method of searching the metadata according to the present invention described with reference to FIG. 11.

[144] The apparatus comprises an input unit 1100 allowing the user to input search conditions, an apparatus for searching metadata 1200 receiving contents, metadata on contents or an index of the metadata, a storage unit 1300 storing therein the received contents, the metadata on the contents or the index of the metadata, a control unit 1400 determining location information of the metadata corresponding to the field of the search conditions input from the input unit 1100, searching the key containing the code predetermined as location information, where at least a part of the location information is defined as the predetermined code, and extracting the concerned metadata by use of the searched key, and an output unit 1500 outputting the result of the search by the control unit 1400.

[145] The control unit 1400 compares the search conditions input from the input unit 1100 with the value of the key contained in the metadata index stored in the storage unit 1300.

[146] Among the steps of searching the metadata according to one embodiment of the present invention, the step of determining location information of the field of the input search conditions within the metadata (S1210), the step of searching the key containing the code predetermined as location information, where at least a part of the location information is defined as the predetermined code (S1300), and the step of extracting the concerned metadata by use of the searched key (S1400) are performed in the control unit 1400. Description of these steps have been described with reference to FIG. 12.

[147] The present invention proposes an index structure providing a simplified indexing for metadata fragments to search the metadata fragments in a quick manner, under the environment wherein the metadata is structured on a fragment basis, a method for searching the index information, and an apparatus for searching the index information.

[148] According to the present invention, quick search for metadata is available and overhead to the apparatus for searching metadata is reduced, thereby shortening the searching time and increasing the efficiency of the apparatus for searching metadata.

[149] Although the present invention has been described in connection with the preferred embodiment shown in the drawings, it is merely illustrative. It will be understood to those skilled in the art that various modifications and equivalents can be made without departing from the scope and spirit of the invention. Therefore, the scope of the present invention should be defined only by the appended claims.